Department of Informatics

A Comparison of Management of Virtual Machines with z/VM and ESX Server

Master thesis

Christer Opsahl
Oslo University College

May 23, 2007

Abstract

Virtualization and virtual machines are becoming more and more important for businesses. By consolidating many servers to run as virtual machine on a single host companies can save considerable amounts on money. The savings come from the better utilization of the hardware, and by having less hardware that needs maintenance.

There are several products for virtualization, and different methods to acheive the virtualization. This thesis will focus on comparing VMware ESX Server and z/VM. These products are quite different and run on different hardware. The primary focus of the comparison will be on management of the two different products.

Preface

This thesis is the conclusion of the Masters Degree on Network- and System-Administration at Oslo College University.

My interest in virtual machines started when I started on the masters degree. The "Firewalls and intrusion detection" course were utilizing virtual machines to provide a virtual network of machines for all the student groups. Providing this would be very difficult without virtual machines.

During this period I started to use Xen, an open-source solution for virtual machines. I was amazed by the ease of setting up new machines. Instead of cluttering my main Linux installation I could create a new virtual machine when I wanted to test software.

My interest in IBM mainframes started during the last semester of the bachelors degree. IBM were offering a course at Oslo University College, "Supercomputers and virtual operating systems" [1]. The course covered the architecture of the System Z mainframe, the z/OS operating system, and an introduction to z/VM.

This thesis started out with a goal of comparing the performance of z/VM and ESX Server with regards to overhead introduced by the virtualization. After some time it became apparent that to compare the systems it was necessary to compare relevant workloads. During the process of finding relevant workloads, IBM experts recommended that the comparison should be on the management instead of performance. The experts suggested that comparing the performance would be too complicated and time consuming to finish in the given time frame.

This suggestion were followed and the focus was shifted to comparing the management of the two systems instead. This caused the comparison to shift from being quantitative to qualitative, which was a change I did not really want. I feel that the thesis lost a lot of it's "edge", but being able to finish the work was most important.

[1] http://www.iu.hio.no/teaching/materials/MS014A

Acknowledgements

There are a number of people I need to thank for their support during the writing of this thesis. The first and most important person is my fiancée, Sølvi Hansen, who has been a great support for me.

I thank Helge Forberg, Jan Ivar Lauten and Jim Roger Olsen at EDB Business Partner for their help, support and motivation during the writing of the thesis. I also want to thank EDB Business Partner for letting me do this work for them.

Per Fremstad and Kristoffer Stav at IBM have done a great job at pushing me in the right direction before and after the work with the thesis started.

My advisor, Tore Jonassen, have done a good job helping me keeping track and answering my questions. In retrospect I realize I haven't used his help enough.

All of the people responsible for the masters degree. Thank you for creating an interesting programme.

InterMedia AS and my boss Aslak Hougen also deserves thanks. During the work with the thesis I've not been as available as I've wanted to be. Too much pending work have been postponed. I appreciate the understanding for my prioritization.

Contents

Chapter 1

Introduction

Virtual machines is an old concept. Virtual Machines were first defined and implemented several decades ago. The latest years it has become more popular than ever. One of the first virtual machine implementations were CP-67 for the IBM s/360 mainframe. The motivation for implementing it was to allow multiple users access to do work on the mainframe. As the name says, it was implemented in 1967, 40 years ago.

This thesis will cover different virtualization techniques and methods, with the primary focus on virtualization on the IBM System z mainframe (z/VM) and on x86 with VMware ESX server. These two products will be compared with focus on management of the different products. There are of course more platforms and products available. Solaris zones, Xen, MS Virtual PC, KVM (linux) and more. The choice of platforms and products were motivated by the needs of the company this work was done for.

The initial focus of this thesis was to compare the effectiveness of z/VM and VMware ESX Server. During the work it became apparent that the comparison would become complex and time-consuming. IBM experts recommended to change the focus from performance to management. This recommendation was followed. As a consequence of this change of focus, the comparison has changed from quantitative to qualitative. Another consequence is that the comparison have been extended to include additional management tools.

There are not many papers focusing on the management of virtual machines. Most cover techniques to implement virtual machines[1] [2] [3] [4] [5] [6] or performance of virtual machines [7] [8] [9]. The complexity of managing virtual machines is interesting to the persons doing the management, and their managers.

The complexity of managing virtual machines defines the need for documentation, mentoring and resources needed.

1.1 The purpose

The purpose of this work is to determine the complexity of managing the different systems, and to answer the question: “Which of the two systems is the best to work with, with regards to management?”.

1.1.1 What is management?

Before the question can be answered we need an understanding of what management of servers is. By using the lifetime of a normal server as a basis, we can list the different tasks that have to be done.

When a company need a server, they plan the kind of workload it will run and the necessary resources to run the workload before ordering it. The company then orders the server from a vendor. In a virtual environment, this is the same except that the company doesn't order a new server, the new virtual machine is defined in the virtualization software.

A server can be delivered with or without an operating system. A server without an operating system needs to have one installed. This is the same in a virtual environment where all new VMs are without an operating system.

When an operating system is installed and the server is running it's designated workload, there are other tasks that must be done. The performance of the server should be monitored to make sure it performs well enough. If there are performance problems, they have to be solved (e.g. by buying additional hardware). The data on the server must be backed up, and there should be a disaster recovery plan for it. This also applies to virtual machines.

When a server is no longer needed it can be sold or end up in the garbage. With a virtual machine, the server is simply removed from the virtual environment and its resources returned to the environment.

We end up with the following list of tasks related to managing a server:

1. Plan resources
2. Buy new server
3. Install an operating system and applications
4. Monitor performance, solve performance problems
5. Backup data
6. In case of disaster, effectuate disaster recovery
7. When the machine is no longer needed, get rid of it

1.1.2 Criterias for comparison

To be able to compare the two systems we need a set of criterias. One natural criteria is the properties and features of each system. This criteria cover both hardware and software.

The second criteria is how easy the systems are to use. A system should be easy to use, and still not hinder users with great knowledge of the system.

Availability of documentation, and quality of the documentation is the third and last criteria. The availability and quality of documentation greatly influence the system administrators ability to learn the systems, and use them effectively.

1.2 Document organization

This chapter contains the introduction to the work. Chapter 2 cover the background material, e.g. what a virtual machine is and why virtualization is useful. It will also describe different types of virtualization and their differences. Chapter 3 will describe the hardware architectures for z/Architecture and x86, and the software used for virtualization. A more practical view of the systems will be presented in chapter 4. A summarization of the results is done in chapter 5. Chapters 6 and 7 contain a discussion of the results, and a conclusion.

Chapter 2

Background

This chapter covers background information about virtualization.

2.1 What is virtualization?

One definition of the word *virtual* is: *existing in essence or effect though not in actual fact* [1]. By using this definition we can define *virtualization* as *creating an object which exists in essence or effect though not in actual fact*.

An IBM poster from 1978 had the following explanation of virtual memory:

> If it's there and you can see it - it's REAL
> If it's there and you can't see it - it's TRANSPARENT
> If it's not there and you can see it - it's VIRTUAL
> If it's not there and you can't see it - you ERASED IT!

Virtual memory is a very good example of virtualization. In the early days of computers memory were very expensive, while storage (harddrives) were cheaper. As programs grew more complex and needed more memory, it was necessary to give them more memory. Real memory were expensive and it was necessary to find a cheaper solution. The problem was solved by using virtual memory. The memory an application saw was virtualized. By using virtual memory it was possible to use disk-storage to expand the amount of real memory. This expansion were done transparent to the application.

2.2 Virtual Machines

Goldberg[4] defines a virtual machine as "An efficient, isolated duplicate of the real machine". This definition imposes some restrictions on virtual machines. The virtualization must be efficient, inducing only a small amount of overhead. The different virtual machines must be isolated and not allowed to touch each others data. Each virtual machine must be a duplication of the real machine, i.e. an x86 machine cannot present a virtual environment resembling a machine of a different architecture.

[1] http://wordnet.princeton.edu/

Prior to virtual machines techniques for sharing of a computer system included multiprogramming, multiaccessing, multitasking, multiprocessing [10]. With multiprogramming multiple programs were loaded into memory. When a program was finished or stopped, the next program could run. Using multiaccessing multiple users access the system. With multitasking a system can support multiple active processes. Multiprocessing allows multiple processes to be executed concurrently, using more than one processor.

Virtual machines is an extension to these techniques. With virtual machines the hardware resources is shared/split to allow many virtual systems. The aforementioned techniques can be used within a virtual machine.

2.2.1 Processor requirements

The instructions in a processor must behave in a certain way to make virtualization possible. The instructions can be divided into three groups: privileged, sensitive and unprivileged. Privileged instructions have access to the hardware and the machine state. Giving all programs access to privileged instructions gives all programs full control of the machine. Unprivileged instructions are instructions that doesn't access the hardware or machine state. These instructions are "safe", and can be run by any program without compromising the machine.

A privileged instruction is an instruction that can only run in privileged mode. If it is run in unprivileged mode, it will cause a trap, which gives control to some software which will decide how to handle the instruction. Since a virtual machine will run in unprivileged mode (even the kernel of the guest), these traps must be handled by the software providing the virtualization. The software may fake the result of the operation so that the guest will get the result it expects, or it can run the instruction on behalf of the guest.

Unprivileged instructions are instructions that is not privileged, and can run in every mode, both privileged and unprivileged.

Sensitive instructions can cause problems for virtualization. A sensitive instruction can run in both privileged and unprivileged mode, but the result of the instruction depends on the mode it is run in. Instruction i run in privileged mode may return 0, but return 1 if it is run in unprivileged mode.

Grouping the different instructions into different sets, A is the set of all instructions, P is the set of privileged instructions, S is the set of sensitive instructions and U is the set of unprivileged instructions. If S is a subset of P, virtualization is easily possible. However, when S is not a subset of P, virtualization is more difficult. x86 processors prior to getting hardware support for virtualization have several instructions where the set of sensitive instructions is not a subset of the privileged instructions [11].

2.2.2 Virtual Machine Monitor

Virtualization at the hardware layer is done between the hardware and the guest virtual machines. This virtualization is done by having a small layer of software between the hardware and the virtual machines. This is called a Virtual Machine Monitor (VMM).

Popek and Goldberg describes the following properties of a Virtual Machine Montitor (VMM)[4]:

- "Provides an environment for programs which is essentially identical with the original machine"
- "Programs run show at worst only minor decreases in speed"
- "Is in complete control of system resources."

These properties implies that programs running in a virtual machine must show the same behaviour as when it is running on a physical machine (except timing, and resource availability). "Duplicate" implies that a time- sharing OS can not be classified as a VMM.

Efficiency

The requirements set by Popek and Goldberg [4] requires that "a statistically dominant subset of the virtual processor's instructions be executed directly by the real processor". The instructions that cannot be executed directly on the real processor cause VMM intervention. The VMM can emulate the instruction for the virtual machine. Every VMM intervention causes overhead and to maximize the efficiency, it is necessary to minimize the number of VMM interventions.

2.3 Virtualization

Virtual machines can be implemented in different ways, and at different levels.

2.3.1 Different techniques

There are different ways of providing a virtual machine:

- Full virtualization
- Para-virtualization
- Emulation

With full virtualization, the guest OS runs without modifications and a full virtualization of the hardware is provided to the guest. The virtual machine is a virtualization of the underlying hardware.

It is also possible to modify the guest OS to be aware that it is running in a virtual machine. This is called para-virtualization. By modifying the guest OS, it is possible to make the virtualization more efficient. Para-virtualization also provides a virtualization of the physical hardware.

By using emulation, the underlying hardware is not virtualized. By using emulation the environment of a virtual machine can be an Amiga, or any other emulated hardware.

2.3.2 Different layers

The virtual environment can be provided at different layers in a server. The layer used to provide the virtual environment influence the performance of the virtualization.

- Between hardware and operating system
- Between operating system and application
- Between application and application

Providing the virtual environment between the hardware and operating system requires a Virtual Machine Monitor. The purpose of the VMM is to provide isolation, scheduling, management of resources and to make sure that the virtual machines can run properly. In terms of performance, isolation and resource control, virtualization at this layer is advantagous. One problem with this layer is that multiple guest operating systems are run at the same time. This consumes more resources than virtualization at the OS level [1].

Operating system level. Support for virtualization is implemented in the operating system. The OS is responsible for providing the virtual machine environments. Examples of virtualization at this layer include Linux VServer, *BSD jails and more. By using this layer a some resources are saved. The isolation of the servers is done by the OS. Only one OS kernel needs to be running and there is only one OS responsible for paging. The VMs can get access to hardware via the host OS's drivers. Linux-VServer, OpenVZ and FreeBSD jails are examples of software providing virtualization at the OS-level.

Virtualization at application level is done in an application. Examples of virtualization at this level includes the Java Virtual Machine, which is a virtual machine designed to run java bytecode. Emulators are also often running at this layer.

2.4 History of virtualization

IBM

Cambridge University started using CP-40 in 1966. It was running on a modified System/360 model 40 (a dynamic address translation (DAT) device was added). With System/360 Model 67 a DAT device were included. CP-67 were written to facilitate this device. VM/370 for System/370 was released in 1972. Virtual memory for System/370 were announced at the same time.

In 1987 IBM introduced logical partitioning on their mainframes. A logical partition is in effect a virtual machine. Until the z990 mainframe, using LPARs was optional. The z990 and the newer z9 cannot run without logical partitions [13].

System/370-XA (Extended Architecture) was introduced in 1983. Among the enhancements were 31-bit address spaces and the interpretive-execution achitecture. The host/Control Program (CP) can put the machine in interpretive-execution mode by issuing the `start interpretive execution` (SIE) instruction. In this mode the machine can support different architectures, S/370, S/370-XA, ESA/370, ESA/390 or VM Data Spaces mode. Special hardware allows the machine to efficiently interpret

functions. Most of the functions are handled by hardware, except some which are handled by the control program using simulation.

The history of IBM and VM is covered in great depth by Melinda Varian [14].

More stuff here.

2.5 Why use virtual machines?

One of the biggest benefits of virtualization is consolidation of otherwise underutilized servers onto fewer physical servers. Unix and Windows servers often have poor processor utilization. A server must be able to handle the load at all times. The load on a server is usually not at the average at all times, but varies with the time of day and the type of server. Since the load varies, traffic peaks occur. The server must be able to handle the peaks.

Using virtual machines can lead to better utilization of the hardware if servers with loads that doesn't coincide is migrated to virtual machines. All physical machines need some resources, like physical space, power, network connection, cooling.

2.5.1 Advantages

- Server consolidation (less hardware)
- Better utilization (many servers idle much)
- Less maintenance (less hardware, and people)
- Hardware independence (makes migration possible)
- Better availability (migration of virtual machines)
- Less cabling
- Reduced physical space
- Reduced power consumption

2.5.2 Disadvantages

- Many eggs in the same basket (hardware failure can take down many servers)
- Not all workloads are suitable to be run in a virtual machine

2.6 Criterias for effective virtualization

Not all workloads will run effectively in a virtual environment. One of the goals of virtualization is to increase the utilization of the physical resources. The best candidates for virtualization are machines that are underutilized, or heavily utilized in only a small fraction of the time.

No matter how much money is spent on physical machines, the amount of resources will always be finite. In other words, the amount of physical resources is

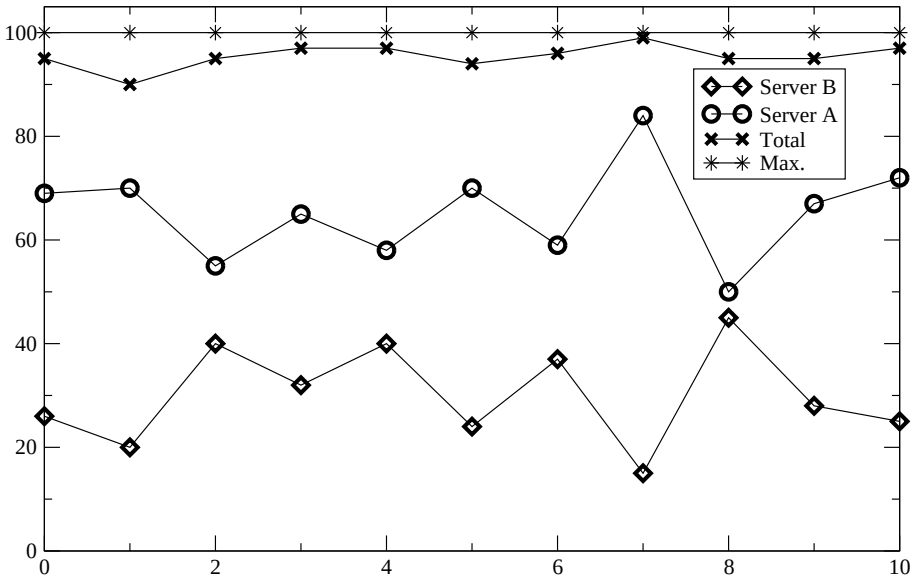

Figure 2.1: Workloads suitable for virtual environments

always a limiting factor. This means that it is necessary to plan the number of virtual machines and what kind of workload they are running. In general, it is advantageous to combine multiple virtual machines with low utilization, or virtual machines on which the high utilization is not coinciding with regards to time. A higher average utilization is wanted. At the same time the system needs to be able to handle peaks in utilization in a good way.

Figure 2.1, and 2.2 on the next page show the workload of two different servers. Server A and Server B have different workloads. The maximum workload the physical machine can handle is 100%. The total workload of both servers is also shown. As we can see from figure 2.1, these two servers fit into a single physical machine.

Figure 2.2 on the facing page shows a situation where the workloads do not fit very well. Because both servers are using significant amount of resources at the same time, they have to compete for the system resources. These servers are not suitable to run as virtual machines on the same physical machine.

A scenario with coinciding high utilization leads to a very high utilization in periods of time. It also causes latency to rise because multiple virtual machines are contending for the resources at the same time.

One example of a problematic workload is misconfigured Linux guests. By default Linux runs some jobs at a specific time every day (e.g. updatedb). Considering a scenario with 100 virtual machines running a job at the same time, problems will occur. Using updatedb as an example. Updatedb indexes all the files in the filesystems that belongs to the virtual machine. Running 100 very IO intensive and possibly long-running jobs at the same time *will* cause problems. This problem is described by Turk and Bausch [15].

Monitoring may also be a problem for virtual machines. Like all other programs a monitoring agent consumes resources. On a real machine with lots of spare capacity, it is usually not a problem if the monitoring agent uses 5 percent of a processors capacity.

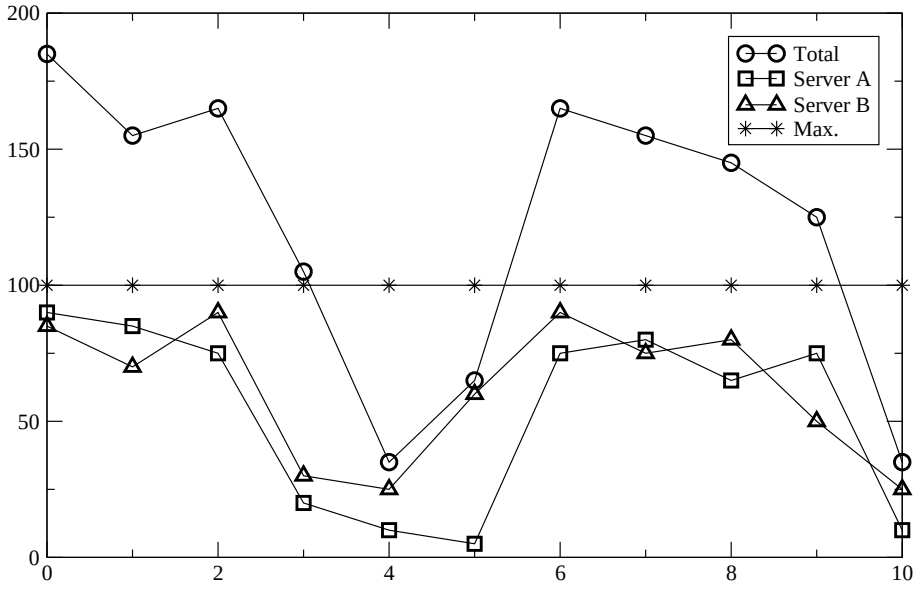

Figure 2.2: Workloads unsuitable for virtual environments

With many virtual machines on the same host this quickly becomes a problem. With 20 guests, a full cpu will be used for monitoring. When choosing the monitoring agent it is therefore important to choose a lightweight one.

Chapter 3

Theoretical comparison

This will be a theoretical comparison of the different virtualization techniques, the relevant hardware and the relevant software. It is necessary to describe both hardware, software and the virtualization techniques to get a full picture.

The first section will be a description of the virtualization techniques, the second will be a comparison of the hardware, and the third will be a comparison of the software. In the fourth section the hardware support for virtualization will be described. The following sections describe the VM software, needed infrastructure and general advantages and weaknesses of each system.

3.1 Virtualization techniques

The two systems use different techniques to provide the virtual environment and to run the guests.

3.1.1 ESX Server

The x86 platform is not ideal for virtualization. One of the problems is that the hardware instructions does not behave properly [7] according to the requirements for virtual machines [4]. The newly introduced hardware support for virtualization removes this specific problem.

On the x86 architecture without hardware support, the running program can find out what privilege level it is running in and some privileged instructions do not trap when running in user-mode. This is a major problem because privileged instructions must be handled by the VMM. This problem can be solved by using binary translation. VMware uses binary translation to avoid this problem. The binary translation changes the problematic instructions while the program is running to avoid the problems.

Binary translation use cycles, but the translation can also reduce the amount of cycles used. Translating a privileged instruction so that traps are avoided can reduce the amount of cycles used. Adams and Agesen compared the amount of cycles spent on the `rdtsc` instructions on a Pentium 4 [7]. Using trap and emulate it used 2030 cycles, while using translated code it used 216 cycles.

By using adaptive binary translation, the number of traps caused by non-privileged instructions are dramatically reduced. Instructions that trap frequently are translated

to avoid the trap.

3.1.2 z/VM

The z/Architecture has very good support for virtualization. The support for virtual machines on IBM mainframes started with CP-67 in 1967. IBM have since then continued development of both software and hardware to support virtual machines efficiently.

IBM introduced interpretive execution with System/370-XA[16]. It has later been extended[17]. The *start interpretive execution* command changes the machine mode to "interpretive-execution". A state description describes the state and architecture for the VM. The hardware allows interpretation to run at speeds close to the native speed. Most instructions are run without intervention, but when intervention is needed, the state description is updated and the execution is returned to the control program. The problematic instructions are then simulated by CP. CP can control the conditions which causes interception by setting the *interception control bits* in the state description for the VM. SIE can be used to virtualize SIE, but the performace is degraded when more than two levels of SIE is used.

A OS sees it's storage as one continuous block. This does not necessarily represent the actual location of the storage. The block the OS sees can be a collection of memory pages scattered in the real memory. Before the machine can access the memory a guest addresses, it needs to translate the address into the physical address. This dynamic address translation (DAT) is performed by hardware. If the OS is replaced with a hypervisor (CP), a second level of address translation is introduced. The VM's address must be translated to the hypervisor's address which must be translated to the physical address. This translation is also done in hardware.

3.2 Hardware

It is obvious that the hardware of a mainframe and an x86 machine is vastly different. In essence both are computers, but their architectures are different. The price tag also indicates that they are very different. A cheap x86 machine can be bought for a couple of thousand NOK, while the prices for the z9 mainframe starts around 1 million NOK.

3.2.1 General info

Computers need processors, memory, storage and interfaces for the operators. Although these components are needed in all computers, there are many different versions of each component. The components are also interconnected differently.

Observing the z9 mainframe and an x86 server from a birds eye point of view, they are the same. They both have processors that controls them, both have memory and storage for storing data, and they both have a interface for operators. Comparing with this amount of abstraction does us no good. A more detailed comparison will be done in the following sections.

x86

The x86 architecture is one of the most commonly used architectures. It is used in machines ranging from laptops to powerful servers.

Compared to the IBM z9 mainframes, there is not a specific configuration for for x86. There is a variety of different processors available, the amount and type of memory depends on how much memory the motherboard supports. Connections to the network also depends on the motherboard. Some have integrated gigabit network cards, while others don't. IO connections are usually provided by adding special cards. The number and type of cards depend on the motherboard. Some different types: PCI 33MHz, PCI 66MHz, PCI-e and PCI-x. The differences between these is bus speed, bus width and number of devices on each bus.

The processors for x86 are very diverse. Ranging from single core processors with 256KB cache to quad-core processors with 8MB cache. There are a number of different sockets (for connecting the cpu to the motherboard) available. The processors also vary with regards to the memory bus speeds they support. Processors are mainly produced by Intel or AMD. At the time of writing, a popular netshop (komplett.no) offers 47 different x86 processors.

Considering rack-servers, these numbers describe the maximum processors and memory in x86 servers offered (as of may 2007) by some large suppliers:

Dell Max 8 cores and 64 GB memory

HP Max 8 cores and 128GB memory

IBM Max 8 cores and 128GB memory

Sun Max 16 cores and 128GB memory

IBM System z

The IBM System z is IBMs series of mainframes. Mainframes have been used for many decades, and they are still popular. Especially banks, insurance companies, financial organizations, public institutions, big retail-companies, arlines and more companies are using mainframes.

z9 is the newest generation System z. This mainframe supports up to 54 processors and 512 GB memory.

3.2.2 Comparison

This part will contain a comparison of x86 and z9. There is a lot of differences. The comparison will cover: processor, memory, IO, net.

General

IBM has one huge advantage over virtualization solutions for x86. IBM is both creating the software *and* the hardware. By having full control over both hardware and software, it is easier for them to adapt the hardware to support virtualization better.

Virtualization on x86 on the other hand has an advantage with regards to price, and initial cost. It is much easier to buy a 20000 NOK server and expand later, than to buy an z9 up front for more than 1 million NOK.

If a big company definitely know that they will have enough use of a mainframe, it will be easier for them to buy a mainframe than it would be for a smaller company that doesn't necessarily know if they will need it, or if the business will expand (fast) enough.

x86 hardware is generally cheaper, one of the reasons for this is likely that x86 hardware is produced in vastly higher quantities than z9 hardware.

z9

The architecture of the z9 server is different from the x86 architecture. The physical server contains: power-supplies, refrigeration units, CEC cage, IO cage(s), optional batteries and support elements. The CEC cage contains the processors, memory and connections to the IO cages. All components in the z9 server are at least duplicated.

CEC cage The CEC (Central Electronics Complex) cage can host up to four "books". A single book contains up to 16 processors, 128GB memory and 8 MBA (Memory Bus Adapter) cards. The books in the CEC cage is connected in two separate rings. These provide redundant connectivity between the books. Even though the processors and memory is located on different books, a System z server is a symmetric multi processor (SMP).

Processors The processors in a book are located on a single multichip module (MCM). This module contains processor units (PU), system controller (SC), 40 MB L2 cache (SD) and storage control (MSC) chips. There are 8 processor units on each MCM. On all but the largest z9 configuration, 4 of the PUs are single core and the last 4 are dual-core. Each PU has 512KB level 1 cache, 256 KB for data and 256 KB for instructions.

On the largest z9 servers, with 4 books installed, there are 54 available processors (plus 2 spares, and 8 SAP processors). These have a total of 160MB level 2 cache available. The amount of level 2 cache is important when running many virtual machines. A context switch between guests and z/VM cause movement of data between the CPU(s) and memory. Accessing data in level 2 cache is significantly faster than accessing it from RAM.

Official pricing of the normal processors have not been found, but the price for the IFL processors is approximately $95,000 (z9 BC) or $125,000 (z9 EC) USD [18]. The IFL processors can only be used with z/VM and Linux workloads and is cheaper than the normal processors.

Memory The total amount of memory supported is 512GB. The memory is spread between 4 books which can contain 128GB each. Memory can be ordered in 16GB increments.

IO Each book has 8 Memory Bus Adapters, each with two full-duplex 2.7GB/s STI (Self-Timed Interface) connections. The STIs are used to connect to IO. Utilizing all STIs gives a maximum internal IO speed of 172.8 GB/s full duplex.

The z9 have three IO-cages. By default only one is used, but the customer can order more. The IO cages support connections to the disk-systems and network. It also supports cards for Coupling Links and crypto accelleration. Each IO cage has 28 slots. The types of cards and number of cards is decided by the customer.

Disk The disks are not local to the mainframe, but placed in dedicated disk-units. These units are connected to the mainframe via dedicated channels. The z9 can support two different channel types, ESCON and FICON. SCSI is also available with z/VM and Linux. The Ficon Express2 delivers a maximum of 270MB/s per channel. A z9 can support up to 336 of these channels, giving a maximum of 90.720GB/s between the disk-systems and central storage.

Network The network interfaces are located in the IO cage(s). There is a variety of different cards available, the fastest one being OSA Express2 10GbE, which can support up to 24 ports.

There is also support for local networks within the mainframe. Up to 16 hipersockets, networks between LPARs, is supported.

x86

As mentioned earlier, the x86 architecture servers are more diverse than the System z mainframe. There exists a lot of different possible configurations. There are many different types of processors and memory available.

Processor x86 processors can be 32-bit or 64 bit. Most are manufactured by AMD or Intel. Both of these companies have a wide range of different processors. Ranging from cheap desktop processors to processors meant for servers and/or large clusters. Both companies have recently added hardware support for virtualization to their processors.

Memory As with x86 processors there is a variety of different types of memory available. DDR, DDR2. The memory speed depends on the memory bus between the memory and the processor(s), e.g. PC3200 have a 400MHz bus. PC8700 have a much faster bus. The memory doesn't only differ on bus speed, but also on latency and error-checking.

IO Internal storage is often provided via integrated SCSI or IDE on the motherboard. Depending on the size of the server, it can offer zero or more slots for additional cards for storage. With virtual machines one of the advantages is hardware independence. To be truly independent of the hardware the VM is running on, it must also be independent of the storage. Using local disk drives causes dependence since the disks can only be used by one physical server. By using a storage area network (SAN) instead, the storage is available for many servers via a network.

Network Most servers have network cards integrated on the motherboard. It is not uncommon to have two integrated 1 Gbit network cards. Additional cards can be added to the expansion slots.

3.3 Hardware support for virtualization

Virtualization can be done 100% in software, but doing so may be slow. Adjusting the hardware to accomodate to virtualization can give a performance boost for virtualization.

The mainframe has two huge advantages with regards to hardware support for virtualization. The first one is that virtualization was first implemented in the late 60s, and is thus very mature. The second advantage is that IBM controls both the hardware and the software for virtualization on the mainframe. This makes it a lot easier to adjust the two to work together efficiently.

Intel and AMD have recently added hardware support for virtualization in their processors, Intel VT and AMD Pacifica.

3.3.1 z/Architecture

Since virtualization has been used on mainframes for decades, the hardware support is much more mature than on x86.

Logical Partitioning

The System z servers support dividing the available resources (cpu, memory, channel paths) into subpools[19]. These subpools are called logical partitions (LPAR). Each LPAR can run it's own operating system. The separation of the resources into LPARs is done by PR/SM (Processor Resource/Systems Manager). LPARs are an example of hardware virtualization, like z/VM.

A z9 server can be partitioned into 60 logical partitions. The partitions are defined through IOCD/HCD. Addition and removal of the definitions of the logical partitions is not possible without restarting the server, but an LPAR can be defined but not in use. It is thus possible to define more LPARs than necessary, and let the extra LPARs be unused. Doing this removes the need to restart the server when a new LPAR is needed.

Processors can be dedicated to a partition or shared between multiple partitions. Memory is dedicated to a partition, but can be reconfigured with prior planning.

Interpretive Execution

The *Start Interpretive Execution* instruction is implemented in the microcode of a z9 mainframe. SIE have been described earlier in this chapter.

QDIO and Hipersockets

Queued Direct IO allows the IO subsystem to write directly into a virtual machines memory, bypassing PR/SM and z/VM altogether. When a VM wants to send a network packet, it uses a signal adapter instruction (SIGA). The SIGA contains a pointer to the

data it the VM wants to send. The OSA adapter can read data directly from memory, and write data directly to memory.

Hipersockets is an internal network within the mainframe. It is implemented in microcode. The microcode emulates an OSA-express QDIO interface. It's purpose is to provide fast network connectivity between LPARs. Communicating via HiperSockets does not use the IO subsystem or an OSA-express adapter. Hipersockets is also referred to as Internal QDIO.

3.3.2 x86 architecture

Intel and AMD have recently introduced hardware support for virtualization. This support introduced an additional execution mode, the *guest* mode, which has less privileges[7]. This execution mode executes guest code directly. A *virtual machine control blocks* (VMCB) contain the state descriptions of the guests. The VMCB also contain control bits which control the conditions for VMM intervention. When VMM intervention is needed, the machine exits from guest mode, the VMCB is updated and execution is continued in the VMM. The VMCB contains information about why the guest exited. When the VMM has finished, the machine is switched to guest mode again and execution of the guest continued.

Exits from guest mode are expensive, and influence the performance of the virtualization. A guest that never exits will run very fast, while a guest that exits very often will run very slow. Guests doing IO will cause many exits, while guests that only do calculations will cause very efew exits.

The performance of the hardware support for virtualization on x86 is not yet good enough. VMware compared virtualization with hardware and software[7]. The software approach performed better. Their comparison shows that the performance gets better with newer processors.

3.4 VM software

z/VM and VMware ESX Server is the software used in this comparison. z/VM is the virtualization software for the IBM System z servers, and VMware ESX Server is one of VMware's products for virtualization on the x86 platform. ESX Server is a hypervisor. z/VM is the virtualization product for the System z servers. z/VM can be run directly in a logical partition on a System z server, or under another instance of z/VM.

3.4.1 Management

Comparing the management of the two different products is interesting. The complexity of management gives us insight into how much time (and money) must be spent on management tasks. The complexity also shows how much special knowledge is needed. This again shows how easy it will be to replace an administrator (e.g. if the person becomes fatally ill or is otherwise prevented from doing the job). If a person doesn't need much special knowledge to manage a product, it will be relatively easy to replace that person.

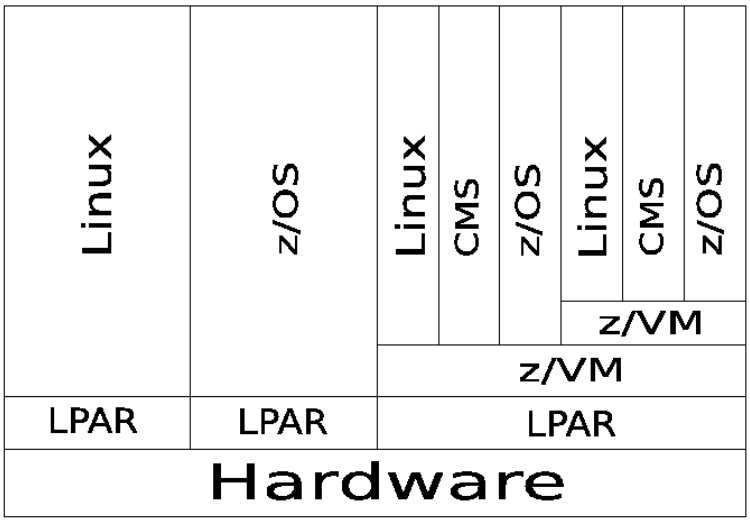

Figure 3.1: Possible ways to partition and use VMs on Series Z

Comparing the complexity of management is difficult to do in a 100% objective way. Some people may find some aspects to be obvious, while others find them to be difficult. This comparison will be performed by the writer of this thesis, and will be as objective as possible. The writer does not have extensive prior knowledge of either product, and should not be biased in any direction. Since his knowledge of both products were on approximately the same level, he should be able to give an objective comparison of the complexity of managing the products. His knowledge of the different hardware platforms is biased to the x86 platform. This reflects the general platform knowledge, since the x86 architecture is much more widely in use than the z/Architecture.

The rest of this thesis will mainly be focused on management.

3.4.2 z/VM

As previously written z/VM is the product for running virtual machines on the System z servers. As the name implies z/VM is written for System z and is not compatible with other hardware architectures. Figure 3.1 the different layers that operating systems can run in.

Hardware

z/VM supports up to 32 processors and 128GB of memory. A z9 mainframe supports more processors and memory than this. To utilize the rest of the resources, additional z/VM installations can run in other LPARs.

Supported Operating Systems

z/VM supports the operating systems that can also be run in a logical partition on the mainframe. This includes: z/OS, OS/390, VSE/ESA, z/VSE, TPF, z/TPF, VM/ESA,

z/VM, z/OS.e and linux [20].

Memory management

z/VM have a hierarchy of storage[21]. This hierarchy contains of main memory, expanded memory and page space on DASD (disk). Main memory is directly available for the programs, and directly addressable. Expanded memory exists in the physical memory, but is only addressable by whole pages. Paging space is storage on disk drives (DASD).

z/VM can use expanded storage as a fast paging device. Paging to physical memory is much faster than paging to disk. VMs execute in main storage, and since not all of their memory pages are used all the time, some pages can be moved to expanded storage or disk. Pages in expanded storage can be moved to main storage or page disks.

Linux will by default use all the available memory. It will use memory to cache data from disk to save disk accesses. This makes sense on a dedicated computer, not using the memory would be a waste of available resources.

z/VM has a minidisk cache which is available for all VMs. If linux VMs are allowed to use large amounts of memory to cache the disk, the value of the minidisk cache diminishes and the waste of memory increases. Since the minidisk cache is shared between the VMs, it is better to use memory on the minidisk cache and reduce the caching as much as possible in the VMs.

z/VM also supports virtual disks (VDISK), which are minidisks which only exists in memory. These are great swapdisks for linux. The virtual disks doesn't use memory until they are used. As long as a VM doesn't swap, no memory will be used by the vdisk.

When configuring z/VM it is necessary to plan for paging. Over-commitment of memory enables more VMs to run. It is generally not recommended to over-commit memory more than 1:2. The level of over-commitment depends on the workloads of the virtual machine. The z/VM paging algorithms are tuned for a hierarchy of storage.

Network

Connecting every virtual machine to the network with a dedicated network card causes a large demand for network cards and connections from the network cards to the external network(s). To solve this z/VM supports multiple types of internal networks.

CTC A Channel To Channel network is a connection between two VMs.

Guest LAN A guest LAN is a simulated network within z/VM. Many VMs can connect to it and communicate directly with each other.

VSWITCH A Virtual Switch is much like a guest LAN, but it also supports connecting to a real network interface (OSA adapter). The VSWITCH can operate on two layers of the IP-stack, layer two and three. With layer three, the VMs address each other with IP-addresses, while they use MAC-addresses when the VSWITCH is in layer 2 mode.

HiperSockets HiperSockets are networks between LPARs. The functionality is implemented in the microcode on the System z servers. HiperSockets functions like QDIO devices. Since HiperSockets are internal in the server, they are also referred to as internal QDIO (iQDIO).

Disk storage

The disk storage in z/VM is provided from disk-systems with channels to the System z server. When z/VM has been allowed access to a disk, it can use it to store VM files, paging and spooling.

3.4.3 VMware ESX Server

ESX Server is one of VMware's product for running virtual machines. It can be run on x86 architectures (both 32bit and 64bit). ESX Server is running directly on the hardware as a hypervisor while VMware Workstation and Server is running on top of an operating system.

VMware ESX Server is now a part of VMware Virtual Infrastructure 3 (VI). It can no longer be ordered separately. VMware VI is available in three editions [22]: Starter, Standard and Enterprise. The Starter edition doesn't support SAN and cannot be used on servers with more than four CPUs and 8 GB of memory. The Standard edition does not have the limitations that the Starter edition has. It also includes Virtual SMP. The Enterprise edition supports more features than the Standard edition: VMotion, High Availability (HA), Distributed Resource Scheduler (DRM) and Consolidated Backup.

To be able to use VMotion, DRS, HA and Consolidated Backup it is necessary to also buy VirtualCenter Management Server.

Hardware

ESX Server supports up to 32 logical processors and 64GB memory. A VM may use up to 4 logical processors and 16GB memory.

Supported Operating Systems

ESX Server supports a larger group of operating systems than z/VM. This group includes Windows, Linux, Solaris and FreeBSD [23].

Features

Virtual Infrastructure has some interesting features. The availability of these features depends on the edition of VI.

VMotion VMotion enables migration of running guests from one host to another while they are running.

High Availability HA monitors the physical and virtual servers. If a physical server goes down, and there is enough resources in the rest of the cluster, HA can start the VMs that were running on the dead host on the other hosts. If it is used with Distributed Resource Scheduler (DRS), DRS will choose the optimal physical server to start the VM on.

Consolidated Backup Consolidated Backup is a centralized backup solution. It supports backing up and restoring entire images, and also files and directories for VMs running Windows. To reduce the network traffic it is connected to the SAN via Fibre Channel.

Distributed Resource Scheduler DRS monitors the usage of resources in a cluster and balance the resource usage across multiple physical servers. Resources can be added or removed from the cluster, and DRS will adjust to that. When a new host is added, some VMs can be moved from the existing hosts and onto the new host to balance the load. When a host is going to be shut off, DRS can migrate all the VMs to other hosts.

Memory management

ESX Server uses different techniques[8] to manage the memory of a physical machine. Each VM is given a reserved amount of memory, and a limit for the maximum amount of memory it can use. When a VM is started it believes that the amount of memory available is equal to the maximum amount it can use. When there is enough memory on the physical machine, the VM gets the maximum amount of memory.

In situations when there is not enough memory to let every VM use the maximum amount they're configured for, it is necessary to reclaim memory from the VMs. One possible way to do this is to let the VMM page out memory. This is a problem, because the VMM does not have good enough information about which pages is the best ones to page out. Paging memory transparently to the VM introduces another problem, double paging. This situation occurs when the VMM have paged out a memory page and the VM decides to page out the same memory page. When the VM tries to page out the page it causes a page-fault in the VMM and cause the page to be paged in from disk. The VM then pages it out to disk again. This technique is not very suitable, but ESX Server can use it if it is necessary.

ESX Server primarily uses a technique called *ballooning* to adjust a VM's available memory. A balloon module is loaded into the guest OS. This module communicates with ESX Server. The purpose of the balloon is to use a variable amount of memory. When ESX Server needs to reclaim memory it instructs the balloon to *inflate* and use some of the VM's memory. The memory pages the balloon uses can then be used by other VMs. When there's enough available memory, ESX Server can tell the balloon to *deflate*, and thus free more memory to the VM. Since the balloon claims memory from OS running in the VM, the OS decides which pages to free. If there's not enough memory that can be freed, the OS decides which pages to page to disk.

ESX Server have an additional technique to optimize memory usage. With multiple operating systems running on the same host, it is likely that there are reduntant pages (pages with identical content). *Transparent page sharing* was introduced with

Disco[2]. Identical memory pages are identified, the guest physical pages are mapped to the same machine page. The machine page is marked copy-on-write so that writes to it causes a generation of a private copy.

ESX Server identifies identical memory pages by creating a hash of the contents of memory pages. If the hash matches a page that is already marked copy-on-write, a byte by byte comparison is done to make sure that the pages are identical. If the hash doesn't match a page that is COW, it is tagged with a hint entry. When a future page matches the marked page, the contents of the marked page is rehashed. If the page has not been changed, the page can be shared. If the page has been modified, the hint mark is removed.

The amount of memory that can be reclaimed by page sharing depends on the operating systems in the VMs and the applications they run. Waldspurger reports of reclaimation of up to 32.9% of the memory on a production deployment of ESX Server in a large company.

Network

Each VM running under ESX Server can have multiple virtual network interface (vNIC) cards defined. These can be connected to a virtual switch (vSwitch) within ESX Server. The virtual switch is connected to one or more real NICs. Using more than one NIC enables load-balancing of traffic and failover. By definind port groups on the vSwitch, a vNIC can be connected to the port group instead of a specific port. The vNICs connected to a port group are in the same layer 2 network, even if they are on different physical servers.

The virtual machines access the vNIC with the device drivers provided by the operating system, or via a VMware optimized device driver.

Disk storage

The disk storage used by ESX Server can be local on the server itself, or located on a Storage Area Network (SAN). Using local storage for the virtual machines make live migration of VMs impossible. Consolidated backup also depend on storage on a SAN.

3.5 Infrastructure

The amount of necessary infrastructure dictates the physical needs for a system. Most system have a set of infrastructure that cannot be omitted. Power, network, physical space are examples of necessary infrastructure. Power and network grows with respect to the number of systems. The physical space used grows with respect to the individual system sizes and the number of systems.

3.5.1 Generic infrastructure

In general both z/VM and ESX Server needs at least this infrastructure to work:

- Power
- Cooling

- Network connections
- Switches (and possibly routers)
- Disk-systems and access to them
- Physical space

However, there are differences in the amount of infrastructure needed.

3.5.2 VMware

ESX Server is generally running on smaller hosts than z/VM. This means that the number of hosts must be increased to be able to run the same amount of virtual machines that z/VM handles. This of course depends on the total number of virtual machines.

Each host must have power, network-connection(s) and connection to the disk-system(s). If there is a large amount of hosts, these connections add up to a significant number.

3.5.3 z/VM

z/VM is running on a System z server which can be scaled to handle quite a bit of load. Scaling the server leads to less hosts, which leads to less cables, power and cooling.

3.6 General advantages and weaknesses

The different platforms are designed differently and have some characteristics which make them better at certain things.

System z servers are very good at processing large amounts of data fast. The amount of memory supported, the amount of disk-space which can be connected to it, and the hardware support for virtualization also makes it good for virtual machines. The z/Architecture is also very efficient at managing memory.

x86 is a more general platform which aims at being good at many aspects. The x86 servers generally have very fast processors, but have constraints with regards to IO.

Chapter 4

Practical comparison

This chapter will cover a practical comparison of z/VM and VMware ESX server. The comparison will not cover administration of disks since that is often done by a dedicated group of people. VMware ESX often use a SAN for storage, while on System z the disks are administrated by a group of people whose responsibility is to administrate the disks. During this comparison it is assumed that the necessary disks are available when they are needed.

4.1 Methodology

The experiment will be done by comparing how to do the same tasks on the two different systems.

Tasks:

- Installation of z/VM and VMware ESX
- Configuration of z/VM and VMware ESX
- Creation of VM
  - Resource allocation
  - Resource preparation
  - Installation of guest OS
  - Cloning an installation to more VMs
- Management
  - System monitoring
  - Resource monitoring (per VM and whole system)
  - Change of resources
    - Add/remove resources
    - Throttle resource usage
    - Prioritizing the unused resources
  - Backup/restore

- Freeze/hibernate guest and move it (migration)
- Disaster recovery

The interesting metrics for the comparison is how easy the different tasks can be done and how easy it is for a person that have no knowledge of the system to replicate the tasks. Another interesting metric is how much of the tasks that can be automated.

The guest operating system that will be used is Suse Linux Enterprise System. Linux is the only operating system that is supported by both z/VM and ESX Server.

4.2 First impression

The general first impression of the two systems is that they are very different, especially with regards to user-friendliness and how easy it is for a person without the necessary knowledge to become familiar with the systems. For a person having Windows/Linux background, z/VM is very unfamiliar. ESX Server on the other hand is very familiar.

4.2.1 User interface

While VMware provide a nice graphical user interface by default, z/VM uses a text-based interface by default. By judging by only the default user interfaces it seems that VMware ESX is far ahead of z/VM. IBM also have a webinterface for administrating z/VM, but it is not installed by default and it requires some work to set up.

4.2.2 Necessary knowledge

Doing the tasks on VMware requires understanding of concepts. By understanding the concepts of what one are trying to acheive, the process of doing basic tasks is quite intuitive and should be straight-forward. z/VM requires more knowledge of how to operate it, a more thorough understanding of what configuration files are available, how they are related to each other, and what changes are necessary to do a task.

Conversational Monitor System (CMS) is the environment in which many changes are done in z/VM. The filesystems and how to access them is quite different for a person with a Linux/Windows point of view. The editor, *xedit*, is quite good, but it requires some training to become an efficient user of it. This is a problem that is not unique to xedit, but it is common for many text-based editors such as *vi* and *emacs*. In VMware it is possible to set everything up without touching a text-editor.

Aquiring the necessary knowledge

IBM is publicizing a lot of documentation for their systems on the internet. The documentation is free. A lot of the documentation, the redbooks, are structured as a case-study showing how to acheive different tasks. They also cover some theory, such as concepts, but the main content is how to accomplish different tasks.

4.3 Initial setup of ESX Server and z/VM

This section will describe the initial installation and configuration of the two systems.

4.3.1 ESX Server

The installation and configuration of VMware ESX Server is covered in “VMware Infrastructure 3: Install and Configure” [24].

Installation

The installation of ESX Server is done by booting the machine from the installation CD. The user is then given a choice between a graphical mode or text mode for the installation. By the look of the graphical mode, it seems that ESX Server is based on Redhat Enterprise Linux or Fedora. These Linux- distributions both have nice graphical interfaces.

The most important steps in the installation is to properly set up the harddrive and the network. The harddrive can be a local drive, or it can be on a Storage Area Network (SAN). Setting up the network properly is necessary to be able to configure ESX Server with the Virtual Infrastructure Client.

The installation of ESX Server can briefly be described like this:

- Insert CD/DVD
- Boot machine from CD/DVD
- Answer questions in the graphical user interface
- Set up the local disk or eventually the disk that the machine is going to boot from on the SAN.
- Configure the network interface for the service console
- Answer some more questions
- Let the installation finish and reboot the machine

Configuration

When the installation of ESX Server is finished, the server will have a webserver running, which contains a quick start guide and a link to download the Virtual Infrastructure Client. The client is a windows program which connects to ESX Server and lets the user configure the server via a nice graphical interface.

The client allows configuration of the following on the host:

- Memory
- Storage and storage adapters
- Network and network adapters
- Licenses

- Security
- Resources.

4.3.2 z/VM

The installation of z/VM is thoroughly covered in a redbook, IBM z/VM and Linux on IBM System z [25].

Prerequisites

A number of resources must be available for the installation of z/VM. A minimum of 5 disks must be available to install z/VM. 5 disks is enough to install z/VM, but more disks are necessary to install guests. It is also recommended to have more disks assigned to paging.

The z/VM system also should have access to the network. The LPAR needs access to an OSA Express adapter. The necessary information about the OSA adapter is name, device number, device type and network type.

To perform the installation it is also necessary to have access to the hardware management console (HMC). The HCM is a dedicated workstation for hardware management. This console have access to a DVD-drive, so that software can be installed from DVD.

Installation of z/VM

The installation of z/VM can be briefly described like this:

- Access the HMC
- Insert DVD
- Load 520vm.ins from DVD
- IPL ramdisk
- Execute *instplan*
- Attach devices
- Execute *instdvd*
  - Choose disk volumes (remember to format them)
- IPL from disk
  - *ipl cms*
  - *instvm dvd*
  - Install service updates
  - *put2prod*
  - *shutdown reipl*

Configuration of z/VM

After having finished the steps listed to install z/VM, a basic system is installed with the default configuration. This configuration should be changed so that it is correct. This configuration is done manually, and is not very intuitive to a person that doesn't know z/VM.

The first important thing to notice is that a virtual machine is referred to as a *user*. Which means that when you log on to a user in z/VM you are actually connecting to a virtual machine. If the VM is not running already it will be started automatically. In the default configuration of z/VM there are some users that are already defined and started. The passwords match the username and needs to be changed at a later time.

Some of the important default users:

MAINT Main z/VM system admin. Like *root* in linux/unix.

TCPIP The VM responsible for the TCP/IP stack.

AUTOLOG1 Responsible for automatically running commands on z/VM IPL.

TCPMAINT z/VM network administrator.

DTCVSW1/DTCVSW2 Virtual Switch controllers.

The important configuration files:

SYSTEM CONFIG . Contains the system configuration. Settings for system name, virtual switch definitions, features and more. Owned by MAINT.

USER DIRECT . User directory. Definition of virtual machines. Owned by MAINT.

PROFILE XEDIT . Configuration of the text-editor. All users have their own file.

PROFILE EXEC . Commands that are run when CMS is started. Like autoexec.bat on dos. Each user have their own file.

system_id TCPIP . Configuration file for TCPIP. Owned by TCPMAINT.

SYSTEM DTCPARMS . Owned by TCPMAINT.

The important steps in the configuration of z/VM is:

1. Customize the system config (name, vdisk setting, vswitches)
2. Configure TCPIP (can be done via a program/wizard)
3. Set TCPIP to be automatically logged on (started)
4. Remove the automatic logon of SFS
5. Set up FTP-server for transferring files to z/VM
6. Format the pagevolumes and minidisks
7. Update the system config to use the pagevolumes and minidisks
8. (Optional) Rename the z/VM system volumes

Customize system config The customization of the system config is done by logging on the *maint* user and changing the `system config` file. This procedure involves releasing the *0CF1* volume from CP (Control Program), linking it to *maint*, accessing it and changing the file.

1. Log on MAINT
2. *query cpdisk.* Check CP-owned volumes.
3. *cprel a.* Disconnect the a volume.
4. *query cpdisk.* Check that the a volume is no longer connected.
5. *link * cf1 cf1 mr*. Link to the cf1 volume.
6. *acc cf1 f.* "Mount" the cf1 volume as f.
7. *copy system config f systen conforig f(olddate).* Backup the system config.
8. *xedit system config f.* Edit the system config
9. Change systemname, vdisk settings.
10. Add a virtual switch
    (a) Add *define vswitch vsw1 vdev primary_id secondary_id*
    (b) Add *vmlan macprefic 020000*
11. *file.* Save the file.
12. Test changes
    (a) *acc 193 g.* Access the volume with the cpsyntax util
    (b) *cpsyntax system config f.* Check the syntax of the system config file.
13. Disconnect volume, and access it with CP
    (a) *rel f (detach).* "Unmount" the volume mounted at f.
    (b) *cpacc * cf1 a*. Connect to the volume with CP.
    (c) *query cpdisk.* Check that the volume has been connected.

Configure TCPIP The configuration of TCPIP can be done by issuing the *ipwizard* command when logged on as MAINT. The wizard asks questions about the network configuration, sets it up and tests it.

Automatically start TCPIP, remove SFS and set up FTP

1. Log off MAINT, log on AUTOLOG1
2. *access (noprof.* Don't load the profile (doing so would cause a disconnection).
3. *vmlink maint 191*. Link to MAINT's 191 disk.
4. *copy profile xedit z = = a*. Copy the xedit profile.
5. *copy profile exec a = execorig =*. Backup the profile.
6. Remove the lines for Shared File System (VMSERVS, VMSERVR, VMSERVU).
7. Remove the line with address command.
8. Add autolog of TCPIP, *'cp xautolog tcpip'*
9. Add logoff of AUTOLOG1, *'cp logoff*
10. Log off AUTLOG1, log on TCPMAINT.
11. Rename *profile tcpip* to *"zVM sysid" tcpip*.
    (a) *acc 198 d.* Access the 198 volume where the file is located
    (b) *copy profile tcpip d = tcpiorig =*. Back up the file
    (c) *rename profile tcpip d "zVM system id" = =*. Rename the file.
12. Activate FTP
    (a) *xedit systemid tcpip d.* Open file
    (b) Add:
        AUTOLOG FTPSERVE 0 ENDAUTOLOG
    (c) Remove the semicolons in front of ports 20 and 21.

Set up pagevolumes and minidisks

1. *attach volumeid.* Repeat for every volume id.
2. *cpformat from-id to-id as page*. Format the page volumes.
3. *cpformat from-id to-id as perm*. Format the minidisks.

Use the new pagevolumes and minidisks When the format of the page volumes and minidisks is finished, the *system config* file have to be updated. The changes are made in the section for page volumes and minidisks. The procedure to do this is described in the "Customize system config" paragraph.

Rename z/VM system volumes Renaming the system volumes is necessary if there is more than one z/VM system that have access to the disks which contains the system disks for one of them. The documentation for this task is available[25], and will not be covered here.

4.4 Creation of virtual machines

Installing new virtual machines is one of the important tasks when managing virtual machines. Installing VMs manually can take some time. When installing a new server, it is often desirable that the new VM ends up in a predefined state. This state can include customizations to the operating system and addition or removal of specific software.

Installing manually increases the possibility of human errors. Even though the installation process may be thoroughly documented, a human can skip steps, do typos etc. Automating the installation reduces the possibilities of these errors.

The automatic installation can be done in at least two ways. The first way is to do a normal installation where the configuration is already defined in a file so that the installation is finished without needing user intervention. Examples of such installations is Kickstart, Fully Automatic Installation and Jumpstart. This kind of installation causes all programs to be fetched, unpacked and configured. This can use a lot of resources and take time.

The second way is to create a master image of a VM in the predefined state. This image can be used to be cloned to the new server. This new server can then be customized (change hostname, network settings etc.). By cloning a VM the steps for fetching and unpacking packages, and configuring them can be skipped.

This section is divided into to main parts. The first covers a normal installation of a new VM. The second covers installation by cloning an existing VM, or template.

4.4.1 Normal installation of a virtual machine

After z/VM or ESX Server has been installed there are no virtual machines installed. Because of this we cannot install new virtual machines by cloning existing ones.

z/VM

Short outline:

1. Log on to MAINT
2. Set up the user ident (VM), assign minidisks.
3. IPL CMS
4. Punch the necessary boot files to CMS
5. IPL boot-files.
6. Set up network
7. Continue installation via VNC.

Installing the first linux virtual machine in z/VM is a complicated task.

Prerequisites Some resources must be available prior to the installation of a virtual machine.

Disk storage The VM's files must be stored somewhere.

Memory The VM must have a sufficient amount of memory to run.

CPU z/VM must have enough free CPU resources so the VM can be run.

In addition to these it is necessary to have the installation files available via the network. Since this is the first guest that is installed, these files must be available via the external network.

Resource allocation Disk storage and memory is allocated in the user direct file. Before defining linux users it is recommended to define a profile for the linux users. This profile contains common configuration statements for the linux users. The profile should define the machine-type, cpu definitions, network connections and access to common minidisks.

The important definitions in the profile:

```
PROFILE LNXDFLT
  IPL CMS
  MACHINE ESA 4
  CPU 00 BASE
  CPU 01
  NICDEF 600 TYPE QDIO LAN SYSTEM VSW1
  LINK LNXMAINT 192 191 RR
  LINK TCPMAIN 592 592 RR
```

These definitions sets the linux users to start CMS on logon, use a ESA machine type. Two processors and a virtual NIC is defined (connected to the VSW1 virtual switch). Read access to some disks are also defined.

The definition of the linux user can look like this:

```
USER LINUX PASSW 256M 1G G
 INCLUDE LNXDFLT
 OPTION APPLMON
 MDISK 100 3390 0001 3038 <VMA781> MR PASSW PASSW PASSW
 (and more minidisk definitions)
```

When the linux user is defined, it is activated in z/VM by issuing the *directxa user* command.

Resource preparation To be able to install an operating system in the virtual machine, z/VM must have the boot files available. For Linux guests these files are the Linux kernel and initrd (ramdisk containing device modules). Two more files are needed. An executable script which starts the installation and a parmfile which contains parameters for the installation.

The necessary files are transferred with ftp or via nfs from the external server to lnxmaint's 192 disk. Changing to the correct disk is done by issuing *cd lnxmaint.192*.

The files must be transferred in binary mode and the record format must be fixed 80 byte records. This is set by *bin* and *site fix 80*.

Installation The installation is started by logging on the new linux user. The profile exec located on lnxmaint's 192 disk will be run when the user is logged on. This file does some preparation (definition of a vdisk swap device among other things) and lets the user choose to ipl from disk or not.

Since the virtual machine has not yet been installed, it is not yet possible to ipl from disk. By running the sles9x exec, the necessary boot-files and parameters are loaded and the installation is started. If there are no problems, there will be a message on the console telling the user to use a vncviewer or java-enabled browser to finish the installation.

The rest of the installation is graphical, and similar to a normal installation of linux.

ESX Server

Short outline:

1. Log on the server with the client
2. Define a new VM
3. Mount an iso image in the VM's virtual CD-ROM drive
4. Start the VM
5. Perform installation like on a normal machine
6. Install VMware support tools

Prerequisites As with z/VM, a new VM requires disk storage, memory and cpu resources available. The installation media can be a physical CD, or a CD-image (iso-file). Installing via a CD-image is faster and does not require physical access to the host.

Resource allocation Resources are allocated by defining a new virtual machine.

1. File - new - virtual machine
2. Choose typical
3. Decide a name
4. Decide placement of the VM's files
5. Decide type of guest OS
6. Decide number of virtual processors
7. Decide memory size

8. Decide number of network connections, and what networks they're connected to
9. Decide disk size
10. Review options and finish.

Resource preparation To be able to install via a CD-image it may be necessary to create the image. There are multiple tools for doing this. Most CD-writing software can convert a CD to a CD-image.

Installation Before starting the installation, the installation media must be available and ready. If using a physical CD, the CD must be placed in the CD-rom of the server. If using a CD-image, the image must be connected to the virtual CD-rom of the virtual machine.

The installation is started by powering on the virtual machine. It may be necessary to change the boot-seqeuence in the bios so that the machine tries to boot from CD.

The rest of the installation is performed like on a normal machine, except that the VMware tools should be installed after the normal installation is finished. The VMware tools contains device drivers, and some tools. These enhance the performance of the VM.

4.4.2 Creating additional virtual machines

As we have seen, the process of installing virtual machines can be complicated and time consuming. Doing manual installations is also prone to user errors. Creating template VMs, cloning these and customizing the cloned VMs automatically is a better solution. If there are customizations that needs to be done for all virtual machines, it only needs to be done in the master images. One example of such a customization is the installation of backup software in the virtual machines.

z/VM

Creation of additional VMs in z/VM can of course be done manually as described earlier. More efficient ways of creating additional VMs include cloning existing VMs manually, and using IBM Director.

Cloning manually Cloning existing VMs manually involves configuration and manual work. It also have some prerequisites that needs to be addressed. The first prerequisite is that a master image for the VM is installed. This is the disk containing the files that should be copied to the new VM. Another requisite is that a linux VM with additional system access is installed. This is the VM that does the work of cloning the new VM, the controller.

The cloning starts by adding a user id (VM) to the directory. This user id must be defined with a name, password and the minidisks it have access to. z/VM is then updated so that it knows about the new user id (*directxa user*). The next step is to add necessary information to the AUTOLOG's *user profile* file. Automatic booting of the new VM is added. The new VM must also be given access to a virtual switch.

The script that does the cloning needs some information to run. This information is found in a file on LNXMAINT's d drive. The file is named vmname PARMFILE. The most important information in this file is the network settings.

Before starting the cloning it is necessary to check that the new VM is correctly configured. This is done by logging on the new VM and checking that a NIC is available (this information is automatically shown when logging on), and that the necessary disks are available (*query dasd*).

The actual cloning is done by logging on to the controller VM. This VM have a script named clone.sh in /usr/local/sbin. The script accepts the name of the new VM as a parameter. When started, the script looks up necessary information on LNXMAINT's 192 disk. It then proceeds by copying the disks, mounting the new disk in it's own filesystem and customize the configuration of the new VM. The customization includes generation of new ssh keys, setting the hostname and changing networks settings.

When the clone script is finished, the new VM is started and should be available on the network in less than a minute.

Using IBM Director IBM Director is a tool for systems management. It has support for the System z servers. To use it with z/VM it is necessary to order an extension (z/VM Center). A thorough description of the usage of IBM Director with z/VM is available in a RedBook[1] [26].

IBM Director needs the following components:

- IBM Director Server (with z/VM extension)
- IBM Director Console (with z/VM extension)
- Linux VM (with z/VM Management Access Point (MAP))
- Dirmaint or other supported Directory Manager

The administrator interfaces with Director Server (DirServ) via Director Console (DirCons). The Director Server communicates with the Linux VM, which again communicate with z/VM and does the actual work. The Linux VM communicates with the z/VM Systems Management API, a Directory Manager and CP.

Setting up the Linux VM with MAP requires a bit of work. The configuration will not be covered here, but is available in the RedBook mentioned earlier in the paragraph. The rest of this description assumes that the configuration of IBM Director and the Linux VM is finished and working properly.

The main objects available in the Virtual Server Deployment part of z/VM Center is: virtual servers, virtual server templates, operating systems and operating system templates. The *operating system template* object is an inactive user id with disks containing the files for an operating system. *Operating System* objects contain definitions of the disks that contain it, network devices and other settings. Each Operating System object is associated with a virtual server. A *Virtual Server* object is the same as a Virtual Machine, and consequently also the same as a z/VM user id. A *Virtual Server*

[1] http://www.redbooks.ibm.com

Template object defines templates for virtual servers. These are used when a new virtual server is deployed. The templates contain settings for memory size, number of CPUs, access-level, password and more.

To be able to clone VMs, IBM Director needs a virtual server template and a operating system template. Setting up a new virtual server template is an easy task in which the operator must set some definitions for the virtual servers: pattern for the VMs name, password, the default access class, which prototype to use from the directory, how many processors it can use and the memory sizes. The virtual server template also needs a name and description.

The operating system template is created by defining an operating system on an already installed and working VM. In this process the disk with the operating system must be chosen and network settings set. When the operating system has been defined, it can be used to define an operating system template. The creation of the operating system template let's the user define shared disks (disks which are mounted read-only in all VMs), the name of the template, the user id for the template and a description. The disk pool that is to be used when creation new VMs must also be chosen.

When both templates have been defined it is possible to create new VMs. This can be done in two ways:

1. Create Virtual Server, and then provision an operating system on it

2. Automatically create any number of VMs in a server complex.

Server Complexes allow grouping of VMs. Each group can have different resource allocations. When adding or removing a VM, scripts that modify the VMs can be run.

Conclusion Creation and additional virtual machines is not always an easy task. Installing the additional VMs the same way as the very first VM and cloning is always possible, but the process is somewhat complex and subject to operator errors. Using IBM Director with z/VM Center makes this task much easier, but requires a fully working installation of IBM Director to work. The process of setting up z/VM Center is well documented, but it is complex.

ESX Server

With VMware Virtual Infrastructure there are two ways of cloning virtual machines. The first way is to clone a virtual machine and customize it. The second is to create template virtual machines, clone one of these and then customize the new VM [27].

A template is a master image for a type of virtual machines. A template can be an image of a fully installed guest, with all necessary customizations. A template is created by converting an existing virtual machine to a template.

Creation of a new VM from a template is done by cloning the template to a new virtual machine and customizing the new VM.

Prerequisites To create additional VMs in ESX Server, it is necessary to have either a virtual machine that is suitable for cloning, or a template. As with the normal installation of VMs, disk, memory and cpu resources must be available.

Resource allocation The allocation of resources will be covered in the following paragraphs.

Resource preparation No preparation needs to be done, except to make sure that the necessary resources are available.

Templates A template is a master image of an operating system's filesystem. Templates can be created in three different ways; converting a VM to a template, cloning a VM to a template and cloning an existing template.

It is not possible to make changes directly to the template. The template have to be converted to a virtual machine before making changes, and converted back to a template when the changes are done.

Customization By using the customization feature in the VirtualCenter Client, a new VM can automatically be customized when it is cloned from an existing VM, or when it is cloned from a template. The specification of a customization includes:

- Guest OS
- Guest identification (hostname, owner's name and organization)
- License information (optional)
- Administrator password
- Time-zone
- Network settings (DHCP/static IP address)
- Workgroup/domain (windows only)
- Security ID (windows only)

A guest customization can be applied when cloning a VM and when cloning a template to a new VM.

Cloning a VM Cloning a VM requires that the source VM is powered off. The cloning is done by connecting to the VirtualCenter Server, finding the source VM in the inventory panel and clicking "Clone to New Virtual Machine". This causes a wizard to pop up.

The wizards asks for the following information:

1. VM name, and location
2. Host/cluster on which it is run
3. If cluster, choose host affiliation
4. Which resource pool contain its resources
5. Which datastore will contain its files
6. Customization of the guest

Using a template to create a VM Creating a VM from a template is done by locating the template in the VirtualCenter inventory, right-clicking it and clicking “Deploy virtual machine from this template”. The rest of the creation is identical to when cloning a VM. This is described in the prior paragraph.

4.5 Management

This section of the comparison will cover management of virtual machines. Creation of virtual machines have been covered already and will not be commented on in this section.

4.5.1 Resource monitoring

Monitoring the usage of resources is important to find bottlenecks and causes of performance problems. The main tool for monitoring resources on z/VM is Performance Toolkit which is an optional (but preinstalled) part of z/VM. It is necessary to pay for a license to use it. There are also some free tools included in z/VM as well. On ESX Server the Virtual Infrastructure Client have good support for resource monitoring.

z/VM

Resource and performance monitoring in z/VM [28] can be done with several tools. The CP utilities *INDICATE* and *MONITOR* will be covered, as well as Performance Toolkit.

INDICATE The INDICATE command is available for users with different privilege classes. When it is run by a user in the lowest privilege class, it will only show information for that users VM. For users with higher privilege classes, it will show more information and information for other users as well.

The MONITOR command is not available for all privilege classes. The command starts monitoring performance data.

INDICATE USER This command gives information about a user (VM):

- General information, machine-type, storage sizes
- Number of virtual IO devices
- Usage of paging, pages in real storage, number of page-ins and page-outs
- Usage of expanded storage, paging to/from expanded storage
- Processor usage

INDICATE LOAD System information which shows the contention for resources:

- Total processor usage
- Expanded storage paging and migration rates
- Minidisk cache, read and write rates, percentage of cache-hits.
- Paging rate
- Number of VMs in the dispatch list, the eligible list and the dormant list.
- Usage percentage for each real processor

INDICATE QUEUES Show information about each active VM:

- Transaction class
- Which list, eligible or dispatch
- Status (running, waiting, idle)
- Number of pages in real storage
- Estimate of working set size
- Priority in the eligible and dispatch list
- The processor it is running on

INDICATE I/O Shows information about the I/O activities. Lists the VMs waiting for IO, and the device the last IO request were sent to.

INDICATE PAGING Shows information about the usage of page-devices.

INDICATE SPACES Shows information about memory usage in an address space. Shows usage of real storage, expanded storage and DASD.

Performance Toolkit Performance Toolkit[29] runs in a separate virtual machine. The virtual machine have permissions to monitor performance data from z/VM. Performance Toolkit can be used from the command line, and it also has a web-interface.

Performance Tookit supports live reporting of performance, and also logging of performance data. The operator can choose to work with live data, or stored data. When working with stored data, the interface is the same as when working with live data.

The performance monitoring part of Performance Tookit can monitor a great variety of resources. Only a subset of these will be covered here. The rest is documented in the manual[29].

CPU Load This display shows the CPUs used in z/VM. The display is divided into 6 subparts. The first, CPU Load shows the total CPU load, and a breakdown of how much time has been used in the different states, CP, EMU, WT, SYS, SP, SIC. CP shows time spent in the Control Program, EMU shows time spent in SIE, WT shows time spent in wait state, SYS shows time spent by system services, SP shows time spent spinning a lock, SIC shows percentage of the SIE exits caused because CP had to simulate an instruction. The last field shows the status of the processors. If a processor is dedicated to a VM, the name of the VM is shown.

The second subpart shows general load information. This includes the number of channel commands, IO rate, page rate, expanded storage page rate, the number of privileged instructions simulated by CP per second, the number of diagnose instructions per second, and more.

The third subpart shows information about the queues. This information includes the number of VMs in each queue, the total working set for each queue and more information.

The fourth subpart shows information about transactions. This part shows how many users there are in each transaction group, the transaction rate, average transaction time and internal throughput ratio.

The fifth subpart shows information about users. The number of logged in users, dialed users, active users and queued users. The percentage of users in queue waiting for paging, IO or access to a constrained resource is also shown.

The last subpart shows the heaviest users in the following areas: CPU usage, IO/sec, page rate, pages in real storage, minidisk cache inserts and number of pages in expanded storage.

Storage Utilization This display shows the size and utilization of main storage, expanded storage, minidisk cache, vdisks and page/spool activity.

Information about the main storage include the total size, amount of shared storage, number of locked pages, storage used by trace tables, the amount of memory that is pageable, storage utilization (used by working sets of active users), and tasks waiting for a page frame or a page.

The expanded storage part have information about the total size of the expanded storage, the expanded storage dedicated to virtual machines, how much expanded storage is available to CP, the percentage of the expanded storage used by CP, threshold for migration, allocation rate, average age of expanded storage pages and average age of pages that have been migrated to DASD.

Information about the minidisk cache shows the minimum, maximum, ideal and actual sizes of the minidisk cache in expanded storage and main storage. The minidisk cache read and write hit rate, and the read hit rate and ratio is also shown here.

The virtual disk part contains information about the limits for the amount of memory used for virtual disks by each users, and the total system limit. The number of pages used in main storage, real storage and on DASD is also shown.

Paging and spooling information contains information about paging rates, blocking factor (how many pages read/written in a single operation), the number of IO operations per second used to transfer pages between main storage and paging devices. Information about spool read and write rates are also in this part of the display.

I/O Device This display shows information about the load of the IO devices. All the devices are listed with information about: address, type, label, minidisk links, paths, IO rates, rate of IO saved because of minidisk cache. For each device a list of the time spent in different states is also included. The queue length shows an indication of device contention. The list includes percentages of the device busy time, the operations which were read only, average number of cylinders skipped in each SEEK. Information about throttling include the limit and the rate at which IO were delayed due to throttling.

Channel Load The channel load display shows the different channels, with information about channel id, description, busy-times and a distribution of the channel load.

ESX Server

The VirtualCenter Client can display and log performance data for CPU, Memory, Disk, Network, System and the Distributed Resource Scheduler (DRS). What it can show depends on if it is connected to a VirtualCenter Server or an ESX-Server host. Appendic C in the VMware Basic Administration Guide[27] have tables of what information is available when connected to VirtualCenter Server and ESX-Server.

The performance data can be shown in a graph in the VirtualCenter Client, or exported to excel.

Processor The different processor-related performance information is: CPU usage as percentage, CPU usage in MHz, reserved capacity, time spent in wait state, time spent in ready state, time spent on system processes, extra CPU time and guaranteed CPU time.

Disk IO The performance information for disk IO is: amount of data read and written in the period, number of disk reads and writes in the period and aggregated storage performance statistics.

Memory Usage information about memory includes: percentage of total memory used, zero-pages, amount of granted memory, amount of memory in active use, amount of shared memory, amount of memory that can be swapped, amount of memory that is swapped, amount of memory that is swapped in or swapped out.

4.5.2 Resource control

When a resource problem has been discovered, it is necessary to make changes so that the problem is solved. A range of solutions are possible, from shutting down the VM that causes the problem, prioritizing resources differently, throttling one or more VMs resource usage, to increasing the available resources.

z/VM

z/VM have a range of commands to control resource usage. These are some of the commands from the z/VM CP Command reference [30].

SET SHARE This command changes the resource access priority for a VM. Two settings are available, one for the minimum amount of resources a VM can get, and one for the maximum amount. Both can be set with absolute or relative values. The access to resources can have a hard or soft limit. With a hard limit, the VM will not get any additional resources. With a soft limit, it can get additional resources if there are spare resources available.

SET RESERVED This command reserves a number of pages in real storage for a single VM. This command can be used to make sure that a VM has a certain amount of pages that will not be paged out.

SET QUICKDSP A VM with the quickdsp option set will not have to wait in the eligible queue before it is dispatched. This option is suitable for VMs which have to respond very quickly.

SET THROTTLE This command limits the number of IO operations a guest can initiate to a device. Throttling the IO for a single guest can be useful to make sure that it does not use all of the device's capacity.

DEDICATE / UNDEDICATE The DEDICATE command dedicates a real processor to a VM. The processor will not be shared with other VMs. The UNDEDICATE command removes this dedication.

SET IOPRIORITY This command changes the priority for a VMs IO operations. The priority can be set with an absolute or relative value.

SET SRM STORBUF The SET SRM commands change options for the System Resource Manager. The STORBUF subcommand changes the settings for storage usage for the VMs in the queues. By using values larger than 100, the storage is overcommitted.

ESX Server

Direct resource control in ESX Server is done via the VirtualCenter client. The control of resources is available by opening the properties for a VM.

CPU CPU usage can be controlled by setting the number of MHz a VM is entitled to. The usage of excess CPU-time is adjusted by modifying the number of shares a VM have.

Memory Control of memory is done by defining the amount of memory a VM is entitled to. Both increase and decrease of memory is possible. VMware tools contains a driver that gives ESX Server control of the total available memory for the VM. This is the balloon driver.

When a VM is defined it is given two parameters with regards to memory. The reserved amount of memory, and the maximum amount it can use.

4.6 Backup/restore

Backing up virtual machines can be done in different ways. The backup can be done on disk/partition-level or on file-level. Using an external tool to do the backup without letting the VM know can sometimes work, and sometimes not. The external tool doing the backup doesn't know the state of the programs or data in the VM. One possible problem area is database servers, where the state of the data on disk may not be okey, or in a state where copying the database files will not work when the database server is restarted.

Because of this problem, it is necessary to also do backups on file-level in the VM. This can be done with a backup tool. The VM have full control of the programs it is running and can make sure that the data that is backed up is usable. In the case of a database, it can use the database servers tool to dump the database to disk, and then back up the dump. Doing the same from the "outside" would be difficult.

4.6.1 z/VM

z/VM have multiple tools for backing up data. Backing up on disk-level can be done with DDR and DFDSS. DDR is a manual tool for cloning disks. DFDSS is like DDR, but can be run automatically. It can back up logical and physical volumes.

Backup and Restore Manager for z/VM is a priced feature available from IBM.

4.6.2 ESX Server

A part of VMware Virtual Infrastructure is VMware Consolidated Backup. It is included in the enterprise edition of VMware Infrastructure, and can be ordered as an addon to the other versions.

With Consolidated Backup, a backup proxy is connected to the SAN. This saves network traffic.

4.6.3 Virtual Machine level

Backups from within a virtual machine can be done with a variety of tools. The choice of tools depends on the rest of the backup infrastructure. Most environments usually have an existing backup infrastructure, and the tools used for backing up a VM should use this infrastructure.

The tools used can vary from using rsync to synchronize files from a source to the target, to using special backup solutions like Veritas Backup Exec or IBM Tivoli Storage Manager.

4.7 Migration

Migration of virtual machines between physical machines is a great tool. Being able to move the VMs gives flexibility. When a physical machine needs to be serviced or taken down, the VMs running on it can be migrated to another physical machine. Migration is also useful when adding new physical machines, the running VMs can then be re-balanced on all the physical machines, including the new one(s).

4.7.1 VMware ESX

ESX Server supports migration of virtual machines[27]. This is called VMotion. Without VMotion it can migrate powered off and suspended VMs. With the addition of VMotion, VMs can be migrated while they are still running.

Migration of a powered off VMs can be done between supported CPUs. A powered off VM can be migrated between a host with Intel processor(s) to a host with AMD processor(s). The migration of a powered off or suspended VM follows the following steps:

1. Copy configuration files and disks from source to destination host
2. Register the VM with the destination host
3. Delete the old VM from the source host

Live migration of VMs can only be done between hosts with the same processor type, Intel to Intel and AMD to AMD. The disk storage for the VMs must be located on a SAN. VMotion also requires a dedicated gigabit network for the migration. The same subnets must be available on the two hosts involved in the migration. The live migration follows the following steps:

1. VirtualCenter verifies that the VM is in stable state on the source host
2. The virtual machine's state information is transferred to the destination host
3. The VM continues running on the destination host

With both methods, the VM is not removed from the source host until all data/state information is transferred to the destination host. If an error occurs, the VM will stay on the source host.

4.7.2 z/VM

z/VM does not support live migration of virtual machines at the current time. Work is underway to implement support for this[31]. Suspending virtual machines is not possible under z/VM, so migration of suspended machines isn't possible either.

Migration of powered off VMs is possible. Both VM systems must have the user defined and access to the disks which contain the VMs filesystems. The migration is done manually.

4.8 Disaster recovery

The definition of “disaster” is unclear. What one company call a disaster may not be a disaster for another company. The definition may be set in a company policy, Service Level Agreement or similar. If the company is providing services to other companies, the external companies may have different definitions of what a disaster is.

4.8.1 Hardware failure

Hardware failure on the physical machine(s) running the virtual machines can be a great problem. Since there are multiple virtual machines running on the physical server, a single hardware failure may take down multiple servers.

z9

The z9 server is designed to be resilient to hardware failures [32]. If there are multiple books in a server, removing one while the machine is still running is possible (enough resources must be available in the remaining books).

Power Each book is connected to two Distributed Converter Assemblies. Each of these provide enough power for the whole book. Failure of one DCA does not influence the book it is connected to. A failed DCA can be replaced without taking down the book.

Processor On the z9 server, there are always two spare processors. In the event of a failure, the failed processor is disabled and a spare processor enabled.

Memory The memory in a z9 server support ChipKill. ChipKill can correct multi-bit memory errors. There are also spare memory chips. The memory is continously checked for errors.

IO The z9 supports redundant IO connections. The redundant connections must be via different books. This allows the z9 to keep connections to all IO devices even while a book is being replaced.

The MBA cards can also be replaced while the machine is still running. Redundant IO connections makes sure that the server still have access to the IO devices.

x86

Because x86 hardware is so diverse, it is more difficult to characterize the supported features.

Power Depending on the server, redundant and hot-swappable power supplies may be available.

Processor x86 processors are not hot-swappable.

Memory Different features are available, depending on server and memory type. These features are available on the IBM x260 server:

- Mirroring
- Hot-swap and hot-add.
- ChipKill
- Automatic re-routing to avoid failed chips

Similar features may, or may not, be available on other servers.

IO Depending on the server, hot-swapping of PCI cards can be supported.

4.9 Knowledge

The necessary knowledge to manage the different products differs. While ESX Server offers an intuitive graphical user interface, z/VM offers a not very intuitive command line.

The author feels that the necessary knowledge needed to be able to do the basic tasks in ESX Server is limited to knowing what you are trying to acheive. Using educated guessing a newcomer to the system should be able to understand what is needed to perform basic tasks. This situation is different with z/VM. z/VM requires that the person managing it knows what he is trying to acheive, and also how to acheive it. z/VM is not very intuitive.

Chapter 5

Results

As we have seen in the two previous chapters, the management of z/VM and ESX Server is very different. This chapter contains a summary of:

- Necessary infrastructure
- Software features
- Necessary knowledge for managing the systems
- Documentation

5.1 Necessary infrastructure

z/VM

z/VM has to run on a System z mainframe. As with all servers, a mainframe needs power, cooling and physical space. The mainframe itself occupies 2.5 square metres of space.

Since the storage disks isn't located in the mainframe itself, separate disk systems are necessary. The number of systems and connections to them depend on the requirements for the installation. To provide reduncancy, each disk system should be connected with at least two channels. Tape systems for backup is also necessary. The number and types of network connections also depend on the requirements for the installation.

ESX Server

As ESX is a part of Virtual Infrastructure (VI), the requirements for VI will be described. The necessary infrastructure includes ESX Server hosts, the VirtualCenter Server (and possibly an external database-server), a backup server and a SAN. In addition to these elements, one or more networks are needed (management, vmotion, production).

5.2 Necessary knowledge

Common for both systems is some general knowledge about servers and virtualization. This general knowledge encompasses what virtualization is, what advantages and disadvantages it can give, which workloads are suitable for virtualization and which are unsuitable.

z/VM

The first pieces of knowledge needed to understand IBM mainframes and z/VM is some definitions which are different from the x86-world.

Storage Memory

Central storage Normal memory

Expanded storage A part of the memory which can only be addressed at page level

CP, PU, engine Processor

CP Control Program, the hypervisor in z/VM

DASD Disk storage

Minidisk A partition of a DASD device

VDISK A disk that only exists in memory, “ramdisk”

User ID Virtual Machine

Log on Start a virtual machine

CMS

The default operating system loaded in a VM is Conversational Monitor System (CMS). CMS is a single-user operating system. It’s file system is very different from file systems on linux/unix and windows. It does not have subdirectories, and files are accessed by using a triplet consisting of filename filetype and filemode. The filename and filetype can be eight characters long, and the filemode is one character long. The filemode is used to specify which file system the file is on.

To access a file system, the disk it resides on must be available to the virtual machine. If the file system is not defined for the VM in the user direct file, the disk must first be linked to via another VM. When the disk is available, it can be accessed. Accessing a disk is like mounting a disk in a linux/unix environment.

Certain disks that are used by the control program (CP) must be released from CP before linking and accessing them. This includes the disk containing the user directory.

Memory management

Most guest operating systems are not designed to run in shared environments. They believe that the resources that are available to them are dedicated. This can cause troubles. Linux will by default use most of the available memory. The memory that is left over will be used by buffers and cache.

Since z/VM can cache minidisks, it is a waste of memory to cache the same data in the guests. The memory allocated to a guest should be the minimum it needs, and enough to keep the amount of paging down. The guest can have multiple paging devices with different priorities. The first priority one, can be a virtual disk which resides in memory. The second can be a normal disk drive with minidisk caching turned off. z/VM controls the virtual disks and can page whole or parts of it out to expanded memory or disk.

The goal is to let z/VM control memory usage and not let greedy guests control it. With z/VM 5.3 support for Collaborative Memory Management Assist (CMMA)[33] is introduced. This enables CP and the guest OS to share page status information. The guest OS sets the usage status of each of the pages it have access to. The usage status reflects how important a page is to the guest. CP can then make better decisions when paging to expanded storage or disk. CP sets the residency state for each of the pages. This information can be retrieved by the guest OS. CMMA requires modifications to the guest operating system.

Network

z/VM supports different types of networks. A guest can be connected to physical networks, or to virtual networks. The physical networks include connections via an OSA network adapter, channel-to-channel adapter and HiperSockets. The virtual network connectivity provided by z/VM include guest LANs, virtual switches and point to point connections.

Guests can be directly connected to physical network adapters. This is recommended for guests that have to be connected directly to the network.

HiperSockets are internal networks designed for communication between logical partitions. The functionality is implemented in the microcode in the System z mainframes. Communication is done via the system memory.

Guest LANs are internal networks in z/VM. These networks can not be directly connected to external networks. A router is needed to get access to external networks. A VSWITCH is much like a guest LAN with the exception that it can also connect to external networks. A VSWITCH can operate on layer 2 or 3 of the OSI model. Each VSWITCH can be connected to multiple OSA adapters for redundancy.

Configuration files

Table 5.1 on the following page shows a list of the important configuration files and their location.

Filename	Description	Owned by	Disk
SYSTEM CONFIG	z/VM System configuration	MAINT	0CF1
USER DIRECT	User directory	MAINT	0CF3
PROFILE EXEC	Automatically run on IPL	AUTOLOG1	0191
system_id PROFILE	Definitions for primary TCP/IP stack	TCPMAINT	0198
SYSTEM DTCPARMS	Definitions of additional TCP/IP stacks	TCPMAINT	0198
TCPIP DATA	Definition of DNS and hostname	TCPMAINT	0592

Table 5.1: Important files and their location

ESX Server

To managing ESX Server in Virtual Infrastructure it is necessary to understand some concepts.

Host Physical server, running ESX Server.

Cluster A group of hosts.

Resource pool A collection of resources.

Shares A relative prioritization for resource access.

Port group Cluster-wide network. Like a VLAN.

Memory Management

ESX Server uses three different techniques to optimize the memory usage. The first is ballooning, where a driver is installed in each guest. ESX Server can tell this driver to use the guest's memory forcing it to free memory. The memory pages freed can then be used for another guest.

In situations where the balloon-driver is not efficient enough, ESX Server can also page to disk. This is done transparent to the guest, and can cause double paging.

The third technique ESX Server uses to optimize memory usage is to scan for identical memory pages. Redundant pages are removed, a single page is kept. When one of the guests try to write to the page, a private copy is created for it.

Network

ESX Server supports internal networks. A virtual switch can be connected to multiple physical network cards to provide redundancy and load balancing.

5.3 Documentation

Learning to manage a new system requires training and documentation of the system. By providing good documentation the vendor of a system gives the users of their system a way to increase their knowledge about the system and learn how to use it efficiently. Users who can use the system efficiently will be able to exploit the system's capabilities to a larger degree. This increases the return of investment. On the other

hand, a user is likely to recommend continuing the use of a system he or she has strong knowledge of. Good documentation is thus important for both the users of a system and the providers of it. Users gets more out of the system, while the providers increase the probability that it's users to continue to use their systems.

IBM have very good documentation of their systems and software. The documentation available ranges from a complete documentation of the machine instructions, to documentation of how to install Linux guests in z/VM (and install z/VM itself). In addition to documentation of their systems and software, there are more than 2000 Redbooks[1] available on IBM's webpage. Many of these Redbooks are focused on how to do different tasks in "step-by-step" guidelines.

VMware also have good documentation of their products[2]. Unlike IBM's redbooks, their documentation is more focused on the different tasks for specific roles.

[1] http://www.redbooks.ibm.com
[2] http://www.vmware.com/support/pubs/

Chapter 6

Discussion

This chapter cover a discussion of the results, and how well the systems are suited with regards to the criterias listed in the introduction.

The initial question was: “Which of the two systems is the best to work with, with regards to management?”.

The criterias is:

- The properties and features of the systems
- How easy the systems are to use
- Availability and quality of documentation

6.1 Properties and features

Table 6.1 on the next page show a comparison of the systems.

The first and most obvious difference is the hardware platforms. An x86 server is not like a System z server. One of the large differences is the scalability. A current x86 server scales up to 16 cores, 16MB level 2 cache and 128GB memory. A z9 server scales up to 54 cores, 160MB level 2 cache and 512GB memory. In addition to the 54 processors is 2 spare processors and 8 processors dedicated to handle IO. The connectivity to disk and networks is better on a z9. In a system with three IO cages, a total of 84 IO cards can be installed. There are cards for connections to disk systems, networks, cryptography accellerators and more. Utilizing all the STI-channels between the processor/memory gives a total capacity of 172.8GB/s full duplex.

The z9 mainframe scales a lot better than x86, but VMware has a workaround for this problem. With VMotion and Distributed Resource Scheduler, Virtual Infrastructure can automatically load balance VMs across multiple ESX hosts. While the z9 scales very well by upgrading the capacity, Virtual Infrastructure scales by adding more ESX Server hosts.

6.2 Ease of use

The most obvious difference between z/VM and ESX Server is the default user interfaces. z/VM uses a console, while ESX Server uses a graphical client (VirtualCenter

Property/feature	ESX/VI	z/VM
Hardware platform	x86	System z
Hardware support for virtualization	poor	strong
Max. physical CPUs (cores)	32	32
Max. amount of physical memory	64GB	128 GB
Max. amount of virtual memory per VM	16GB	128GB
Max. number of virtual CPUs per VM	4	
Max. number of VMs	no limit	no limit
Migration of VM (powered off)	yes	yes
Migration of VM (suspended)	yes	no
Migration of VM (live)	yes	no
Performance monitoring (live)	yes	yes
Performance monitoring (historical)	yes	yes
Internal networks	yes	yes

Table 6.1: Properties and features of ESX Server/Virtual Infrastructure and z/VM

Client).

6.2.1 z/VM

For a new user to z/VM, working with the console is not very intuitive. CMS is the operating system, and users without prior knowledge of CMS may find it difficult. The file system resembles a DOS filesystem, but has no subdirectories. To get access to disks, the user has to access the disk. The files on the disk can then be accessed as FILENAME EXTENSION FILEMODE. Filemode is the name of the disk, similar to the drive letters in DOS and Windows.

z/VM have multiple users which control different parts of the system. The few configuration files are located on different disks. Accessing some config files can be difficult (release from CP, link and access disk).

If IBM Director is already installed with z/VM Center, deployment of new machines is an easy task. A preinstalled system set up for cloning of Linux guests is easy to work with too.

A new user will probably have problems with installing and configuring z/VM. As shown in section 4.4 the installation and configuration is a long process with room for mistakes.

6.2.2 ESX Server

The VirtualCenter Client is an intuitive way to access and manage ESX Server and VirtualCenter Server.

Installation of ESX Server on a host should not be difficult. The installation is similar to the installation of RedHat. If the host doesn't use local disks for the system files, setting up the disks and booting from SAN can be difficult.

6.2.3 Conclusion

ESX Server and Virtual Infrastructure is far ahead of z/VM and IBM Director in terms of ease of use.

6.3 Documentation

Both systems have good documentation. The virtualization cookbooks for z/VM are especially good, covering the whole process of installing z/VM, configuring it, installing Linux guests and setting up cloning.

Chapter 7

Conclusion

The main purpose of this thesis was to compare z/VM and ESX Server with focus on management. As the two previous chapters show, there is a great difference between managing the systems. z/VM and System z excels at the hardware level, with scalability and hardware support for virtualization. ESX Server excels at the user interface.

With ESX Server and VirtualCenter it is necessary to know the concepts and what you are trying to acheive, while with z/VM it is also necessary to know *how* you can acheive it. A great advantage for z/VM is that there are good documentation on how to do the different tasks on it.

Comparing two systems with only qualitative metrics is difficult, and prone to be subjective. The writer of this thesis did not have extensive knowledge of either of the systems prior to doing this work. He was left with the impression that managing z/VM is far more difficult to learn than learning to manage ESX Server/VirtualCenter. While z/VM was more difficult to learn, the documentation was very good.

Some absolute recommendations for choice of system can be made. If the virtial machines need to run Windows, FreeBSD or Solaris z/VM is not an option. If z/OS is needed, then ESX Server is not an option.

Deciding which system is the better one not only depend on the criterias covered in this thesis. A company's current infrastructure and knowledge of the employees may favor one system over the other. The answer to the initial question must be: "It depends".

7.1 Future work

An expanded comparison, going more into depth and including more virtualization products can be a project for future work. Extending the comparison to include Xen and Microsoft's hypervisor is an interesting expansion. Expanding the comparison will give a more complete comparison of the virtualization products.

The comparison can also be narrowed down to focus only on one or two subjects. Some subjects which can be chosen is to compare the different products for backing up virtual machines. A comparison of management tools for virtual machines is another interesting subject.

The initial focus on comparing the performance of z/VM and ESX Server were abandoned in this thesis. This is another interesting project for the future.

Bibliography

[1] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA, 2003. ACM Press.

[2] Edouard Bugnion, Scott Devine, Kinshuk Govil, and Mendel Rosenblum. Disco: running commodity operating systems on scalable multiprocessors. *ACM Trans. Comput. Syst.*, 15(4):412–447, 1997.

[3] R. P. Goldberg. Architecture of virtual machines. In *Proceedings of the workshop on virtual computer systems*, pages 74–112, New York, NY, USA, 1973. ACM Press.

[4] Gerald J. Popek and Robert P. Goldberg. Formal requirements for virtualizable third generation architectures. *Commun. ACM*, 17(7):412–421, 1974.

[5] Tal Garfinkel, Ben Pfaff, Jim Chow, Mendel Rosenblum, and Dan Boneh. Terra: a virtual machine-based platform for trusted computing. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 193–206, New York, NY, USA, 2003. ACM Press.

[6] Andrew Whitaker, Marianne Shaw, and Steven D. Gribble. Scale and performance in the denali isolation kernel. *SIGOPS Oper. Syst. Rev.*, 36(SI):195–209, 2002.

[7] Keith Adams and Ole Agesen. A comparison of software and hardware techniques for x86 virtualization. In *ASPLOS-XII: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*, pages 2–13, New York, NY, USA, 2006. ACM Press.

[8] Carl A. Waldspurger. Memory resource management in vmware esx server. *SIGOPS Oper. Syst. Rev.*, 36(SI):181–194, 2002.

[9] Aravind Menon, Jose Renato Santos, Yoshio Turner, G. (John) Janakiraman, and Willy Zwaenepoel. Diagnosing performance overheads in the xen virtual machine environment. In *VEE '05: Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*, pages 13–23, New York, NY, USA, 2005. ACM Press.

[10] Peter J. Denning. Third generation computer systems. *ACM Comput. Surv.*, 3(4):175–216, 1971.

[11] John Scott Robin and Cynthia E. Irvine. Analysis of the intel pentium's ability to support a secure virtual machine monitor. In *Proceedings of the 9th USENIX Security Symposium*, pages 129–144, 2000.

[12] Efrem G. Mallach. On the relationship between virtual machines and emulators. In *Proceedings of the workshop on virtual computer systems*, pages 117–126, New York, NY, USA, 1973. ACM Press.

[13] Jean-Louis Lafitte. 40 years later .... a new engine to handle an operating system infrastructure. *SIGARCH Comput. Archit. News*, 32(4):15–22, 2004.

[14] Melinda Varian. Vm and the vm community: Past, present, and future. Princeton webpage, August 1997. http://www.princeton.edu/ melinda/25paper.pdf.

[15] D. Turk and J. Bausch. Virtual linux servers under z/vm: security, performance, and administration issues. *IBM Syst. J.*, 44(2):341–351, 2005.

[16] Andrie Padegs. System/370 extended architecture: Design considerations. *IBM Journal of Research and Development*, 27(3):198–204, 1983.

[17] D. L. Osisek, K. M. Jackson, and P. H. Gum. Esa/390 interpretive-execution architecture, foundation for vm/esa. *IBM Syst. J.*, 30(1):34–51, 1991.

[18] IBM. Integrated facility for linux (ifl). IBM webpage, May 2007. http://www-03.ibm.com/systems/z/os/linux/ifl.html.

[19] Paul Rogers, Alvaro Salla, and Livio Sousa. *ABCs of z/OS System Programming Volume 10*. IBM corp., Dec 2006. http://www.redbooks.ibm.com/redbooks/pdfs/sg246990.pdf, accessed May 2007.

[20] IBM. *z/VM Version 5 Release 3. Frequently Asked Questions*, April 2007.

[21] Gregory Geiselhart, Laurent Dupin, Deon George, Rob van der Heij, John Langer, Graham Norris, Don Robbins, Barton Robinson, Gregory Sansoni, and Steffen Thoss. *Linux on IBM eServer zSeries and S/390: Performance Measurement and Tuning*. IBM corp., May 2003. http://www.redbooks.ibm.com/redbooks/pdfs/sg246926.pdf, accessed May 2007.

[22] Inc. VMware. Vmware infrastructure 3. pricing, packaging and licensing overview. VMware webpage, 2006. http://www.vmware.com/pdf/vi_pricing.pdf.

[23] Guest operating systems installation guide. VMware webpage, March 2007. http://www.vmware.com/pdf/GuestOS_guide.pdf.

[24] VMware Education Services. *VMware Infrastructure 3: Install and Configure*. VMware, unknown year.

[25] Bradford Hinson Kyle Smith Chris Young Gregory Geiselhart, Yohichi Hara. *IBM z/VM and Linux on IBM System z: Virtualization Cookbook for Red Hat Enterprise Linux 4*. IBM, 2006.

[26] *Managing Linux Guests using IBM Director and z/VM Center*. IBM, May 2007.

[27] Inc. VMware. *Basic System Administration, ESX Server 3.0.1 and Virtual Center 2.0.1*. VMware, Inc., 3145 Porter Drive, Palo Alto, CA 94304, 2006105 edition, 2006. http://www.vmware.com/pdf/vi3_301_201_admin_guide.pdf, accessed May 2007.

[28] *z/VM Performance, version 5 release 2*. IBM corp., 2006. http://publibz.boulder.ibm.com/epubs/pdf/hcsi1b11.pdf, accessed May 2007.

[29] *z/VM Performance Toolkit, version 5 release 2*.

[30] IBM corp. *z/VM: CP Commands and Utilities Reference*, May 2006. http://publibz.boulder.ibm.com/epubs/pdf/hcse4b11.pdf, accessed May 2007.

[31] Romney White. z/vm live guest migration, Feb 2007. http://www.linuxvm.org/present/SHARE108/S9110rw.pdf, accessed May 2007.

[32] Franck Injey, Greg Chambers, Marian Gaparovic, Parwez Hamid, Brian Hatfield, Ken Hewitt, and Dick Jorna an Patrick Kappeler. Ibm system z9 enterprise class technical guide. IBM webpage, Dec 2006. http://www.redbooks.ibm.com/redbooks/pdfs/sg247124.pdf, accessed May 2007.

[33] Martin Schwidefsky, Ray Mansell, Damian Osisek, Hubertus Franke, Himanshu Raj, and JongHyuk Choi. Collaborative memory management in hosted linux environments. IBM webpage. http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux/Collaborative-Memory-Management.pdf, accessed May 2007.